




Doc Code: AP.PRE.REQ

PTO/SB/33 (09-08)  
Approved for use through 10/31/2008. OMB 0651-0031  
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

| PRE-APPEAL BRIEF REQUEST FOR REVIEW   |   | Docket Number (Optional)<br>1801270.00139US1 |  |   |   |  |  |  |  |                                    |                          |
|---|---|--|--|---|---|--|--|--|--|------------------------------------|--------------------------|
|   | Application Number<br>10/802,309-Conf.<br>#3911   | Filed<br>March 17, 2004                      |  |   |   |  |  |  |  |                                    |                          |
|   | First Named Inventor<br>Alexander B. BROWN et al.   |  |  |   |   |  |  |  |  |                                    |                          |
|   | Art Unit<br>2191  | Examiner<br>T. T. Vo                         |  |   |   |  |  |  |  |                                    |                          |
| <p>Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.</p> <p>This request is being filed with a notice of appeal.</p> <p>The review is requested for the reason(s) stated on the attached sheet(s).<br/>Note: No more than five (5) pages may be provided.</p> <p>I am the</p> <table><tr><td><input type="checkbox"/> applicant /inventor.</td><td rowspan="4"><br/>Signature</td></tr><tr><td><input type="checkbox"/> assignee of record of the entire interest.<br/>See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)</td></tr><tr><td><input checked="" type="checkbox"/> attorney or agent of record.<br/>Registration number 42,478</td></tr><tr><td><input type="checkbox"/> attorney or agent acting under 37 CFR 1.34.<br/>Registration number if acting under 37 CFR 1.34. _____</td></tr></table> <table><tr><td>Ronald R. Demsher<br/>Typed or printed name</td></tr><tr><td>(617) 526-6000<br/>Telephone number</td></tr><tr><td>October 27, 2008<br/>Date</td></tr></table> <p>NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.</p> <p><input type="checkbox"/> *Total of 1 forms are submitted.</p> |   |  |  | <input type="checkbox"/> applicant /inventor. | <br>Signature | <input type="checkbox"/> assignee of record of the entire interest.<br>See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96) | <input checked="" type="checkbox"/> attorney or agent of record.<br>Registration number 42,478 | <input type="checkbox"/> attorney or agent acting under 37 CFR 1.34.<br>Registration number if acting under 37 CFR 1.34. _____ | Ronald R. Demsher<br>Typed or printed name | (617) 526-6000<br>Telephone number | October 27, 2008<br>Date |
| <input type="checkbox"/> applicant /inventor.   | <br>Signature |  |  |   |   |  |  |  |  |                                    |                          |
| <input type="checkbox"/> assignee of record of the entire interest.<br>See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)  |   |  |  |   |   |  |  |  |  |                                    |                          |
| <input checked="" type="checkbox"/> attorney or agent of record.<br>Registration number 42,478  |   |  |  |   |   |  |  |  |  |                                    |                          |
| <input type="checkbox"/> attorney or agent acting under 37 CFR 1.34.<br>Registration number if acting under 37 CFR 1.34. _____  |   |  |  |   |   |  |  |  |  |                                    |                          |
| Ronald R. Demsher<br>Typed or printed name  |   |  |  |   |   |  |  |  |  |                                    |                          |
| (617) 526-6000<br>Telephone number  |   |  |  |   |   |  |  |  |  |                                    |                          |
| October 27, 2008<br>Date  |   |  |  |   |   |  |  |  |  |                                    |                          |

Claims 1-26 are pending in this application.

#### SUMMARY OF PROSECUTION HISTORY

On May 5, 2007, the Examiner issued a non-final Office Action rejecting claims 1, 22 and 43 under 35 U.S.C. 102(b) as being anticipated by Muth, "ALTO: A Platform for Object Code Modification," 1999, and claims 1-63 under 35 U.S.C. 102(b) as being anticipated by Hsu, "A Robust Foundation for Binary Translation of X86 Code," 1997.

In a Response filed September 7, 2007, the Applicant provided arguments traversing the rejections from the May 5, 2007 non-final Office Action. No claims were amended in this paper.

On December 17, 2007, the Examiner issued a Final Office Action maintaining his rejections from the May 5, 2007 Office Action with respect to Muth and Hsu.

On April 16, 2008, the Applicant submitted a Response to the Final Office Action of December 17, 2007, along with a Request for Continued Examination (RCE). The Response amended claims 1, 22 and 43.

On July 28, 2008, the Examiner issued a non-final Office Action maintaining his rejections from the previous Office Actions.

#### THE CURRENT STATE OF PROSECUTION

Claims 1, 22 and 43 stand rejected under 35 U.S.C. 102(b) as being anticipated by Muth, "ALTO: A Platform for Object Code Modification," 1999, and claims 1-63 stand rejected under 35 U.S.C. 102(b) as being anticipated by Hsu, "A Robust Foundation for Binary Translation of X86 Code," 1997.

The current state of claim 1 is as follows:

1. A method of grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code, comprising:
  - identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code; and

in response to the identifying, dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event.

Claim 1 recites an explicit link between the dividing step (i.e., dividing a region of memory), and the identifying of self-modifying code events. In particular, the dividing is in response to identifying self-modifying code events. The other independent claims (22 and 43) also recite this explicit link. As is set forth below, neither Muth nor Hsu teaches or suggests any such link.

Muth, "ALTO: A Platform for Object Code Modification," 1999

Muth does not disclose one or more limitations in claims 1, 22 and 43 and thus the rejection under 35 USC § 102(b) is improper and should be withdrawn.

Muth attempts to deal with self-modifying code using techniques that do not involve partitions, and the approach set out in the present independent claims is novel and indeed non-obvious in light of the Muth document.

Muth does not describe grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code. The sections at pages 140 and 40 specifically referred to by the Examiner do not relate to grouping of subject code to account for self-modifying code.

At page 140 of Muth, "partitions" are mentioned in relation to basic blocks in the context of "Procedural Abstraction," as set out in Section 6.3 of Muth. At page 140, Muth describes a process for identifying groups of basic blocks which together define a set of basic blocks having single-entry and single-exit. When identified, as such Muth teaches the use of the procedural abstraction techniques on the functions of the groups of basic blocks, not just on individual basic blocks. Self-modifying code is not taken account of by such grouping.

Muth describes ALTO's interaction with self modifying code on page 40. Muth teaches that some self-modification events cannot be handled correctly in the ALTO program (page 40, lines 5-6). Muth teaches that the ALTO program is not applied to self modifying code events,

and that the self-modifying code is allowed to operate without ALTO's intervention, being treated by ALTO as an unknown function (page 40, lines 12-14).

Muth does not describe "identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code". The example in Figure 2(a) featuring a self-modifying code event is explicitly stated as an example of why self-modifying code is not handled in ALTO (see page 43, lines 6-13).

The passages of Muth the Examiner cites at page 41 and 135 are irrelevant to the claimed step of "in response to the identifying, dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event".

Page 41 describes the process of address translation which is performed by the ALTO program when it re-writes a binary, and in these passages there is no mention of the action being taken in response to the identifying of self-modifying code events in the subject code. As set forth above, the example in Figure 2(a) of Muth, which features a self-modifying code event, will not work in ALTO (see page 43, lines 6-13).

Page 135 describes grouping basic blocks into partitions of similar blocks as a step in producing identical basic blocks. This description has nothing to do with self-modifying code events.

The method, medium and combination claimed in claims 1, 22 and 43, respectively, recite an explicit link between the dividing step and the identifying of self-modifying code events. By referring to the portions of Muth that deal with self-modifying code events, it is clear that the system of Muth cannot be used in any way to correctly account for self-modifying subject code. Nor indeed can the compilation techniques referred to in general terms by the Examiner. Therefore Muth does not teach all of the limitations of claims 1, 22 and 43, so those claims should be allowable.

Hsu, "A Robust Foundation for Binary Translation of X86 Code," 1997

Hsu "A Robust Foundation for Binary Translation of X86 Code" does not disclose one or more limitations in claims 1, 22 and 43 and thus the rejection under 35 USC § 102(b) is improper

and should be withdrawn. Each of the remaining claims depends from one of the allowable independent claims, so the rejections of those claims should also be withdrawn.

Hsu attempts to deal with self-modifying code using techniques that do not involve partitions, so the approach set out in the present independent claims is novel and indeed non-obvious in light of the Hsu document.

Hsu does not disclose "a method of grouping subject code during a translation of subject code into translated target code to account for self-modifying subject code, comprising identifying self-modifying code events in said subject code during translation of subject code into translated code and also during subsequent execution of translated code".

The passage referred to on page 3 of Hsu makes it clear that although self-modifying code is detected, the detection of self-modifying code is performed "to prevent the binary translator from translating self-modifying code" (page 3, lines 16-17). Thus in Hsu, there is no grouping of subject code during a translation of subject code to account for the self-modifying code. There is no link between the identified self-modifying code and the translation, nor is there a link between the self-modifying code and the subsequent execution of translated code.

The passages of Hsu at pages 26 and 56 are irrelevant to the step of "in response to the identifying, dividing a region of memory containing said subject code into at least one subject instruction group of subject addresses, wherein each subject instruction group includes one or more ranges of subject code addresses in said memory which are affected by a respective self-modifying code event," as recited in the independent claims.

The claimed method, medium and combination have an explicit link between the dividing step and the identifying of self-modifying code events. By referring to the portions of Hsu that deal with self-modifying code events it is clear that the system of Hsu cannot be used in any way to correctly account for self-modifying subject code. Nor indeed can the compilation techniques referred to in general terms by the Examiner. Therefore Hsu does not teach all of the limitations of independent claims 1, 22 and 43, so those claims should be allowable.

Similar considerations apply to each of the rejected dependent claims, which are likewise not anticipated by Hsu. Those dependent claims should be allowable as being dependent on an allowable base claim.